



Flask 기초

04 SQL Alchemy



목차

01. SQL Alchemy와 ORM

02. DB와 Model

03. Query

수강목표

ORM을 이해할 수 있다.

SQLAlchemy를 사용한 ORM을 이해하고 사용할 수 있습니다.

ORM을 사용한 모델을 작성할 수 있다.

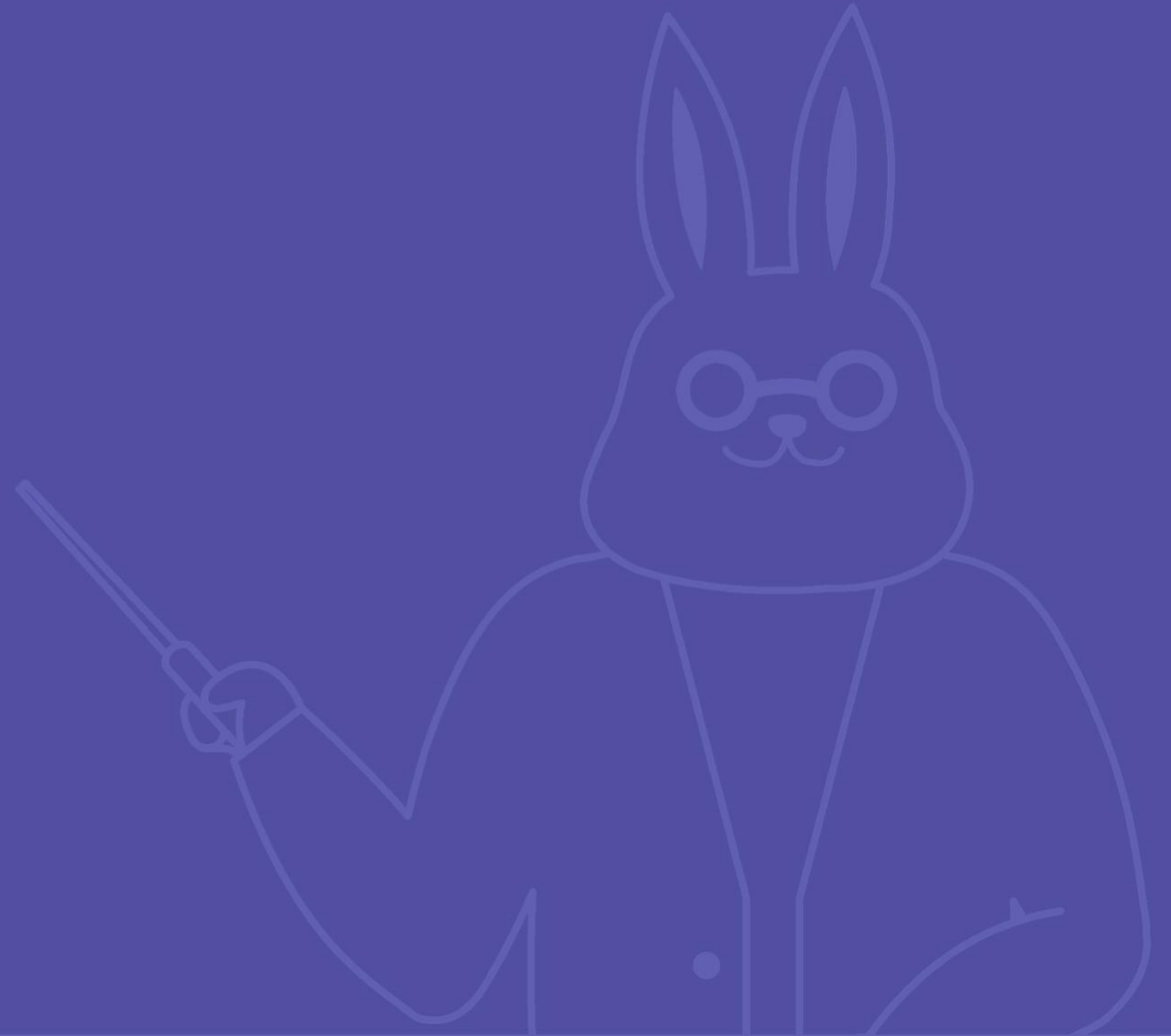
이해한 ORM을 바탕으로 우리가 원하는 데이터베이스 모델을 만들 수 있습니다.

ORM의 쿼리를 이해할 수 있다.

ORM의 쿼리를 이해하고 상황과 조건에 맞게 사용할 수 있습니다.

01

SQLAlchemy와 ORM



✓ ORM이란?

데이터베이스에 객체를 통해 접근하는 방법을 **ORM**(Object Relational Mapping, 객체 관계 매핑)이라고 합니다.

ORM은 **SQL** 지의어가 없어도 데이터베이스를 다룰 수 있도록 도와줍니다.

✓ SQL 쿼리 vs ORM

엘리스 테이블

name	age
엘리스	15
도도	16
체셔	17

SQL 쿼리문	INSERT INTO 엘리스 (멤버, 나이) VALUES ('여왕', '18');
ORM	member1 = Member() member1.name = '여왕' member1.age='18' db.session.add(member1) db.session.commit()

✓ ORM의 장점

DB에 대한 큰 고민 없이, **데이터베이스를 코드로 다룰 수 있습니다.**

테이블 구조가 변경될 때, **ORM 모델만 수정**하면 됩니다.

코드로 작성하기 때문에 **쿼리를 직관적으로 이해할 수 있습니다.**

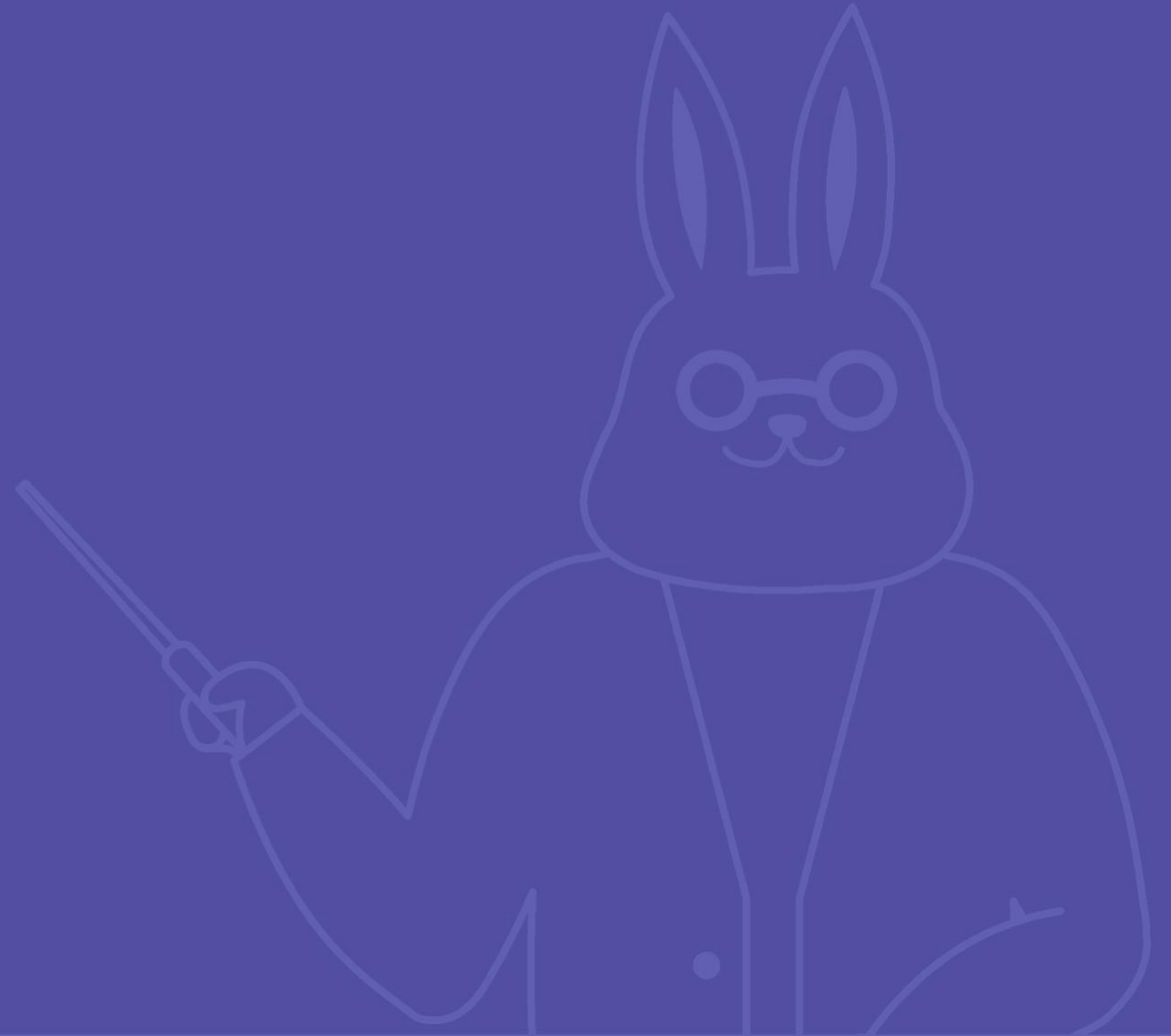
✓ SQL Alchemy란?

파이썬 ORM 라이브러리

→ 파이썬 코드에서 **Database와 연결**하기 위해 사용할 수 있는 라이브러리입니다.

02

DB와 Model



✓ DB와 Model

models.py

```
member1 = Member(name='여왕', age='18')  
db.session.add(member1)
```

- 위의 코드에서 Member는 파이썬 클래스이며, DB의 Member 테이블과 매핑하여 사용합니다.
- DB의 테이블과 매핑되는 클래스를 모델이라고 합니다.

✓ ORM Model

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

- __tablename__ 은 데이터베이스 테이블을 명시해 줍니다.
- id, name, age는 데이터베이스 테이블의 컬럼을 명시해 줍니다.
- 해당하는 데이터베이스를 다룰 때, Members 클래스로 접근합니다.

```
members = Members()
```

✓ ORM Model

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

members = Members()



members.id



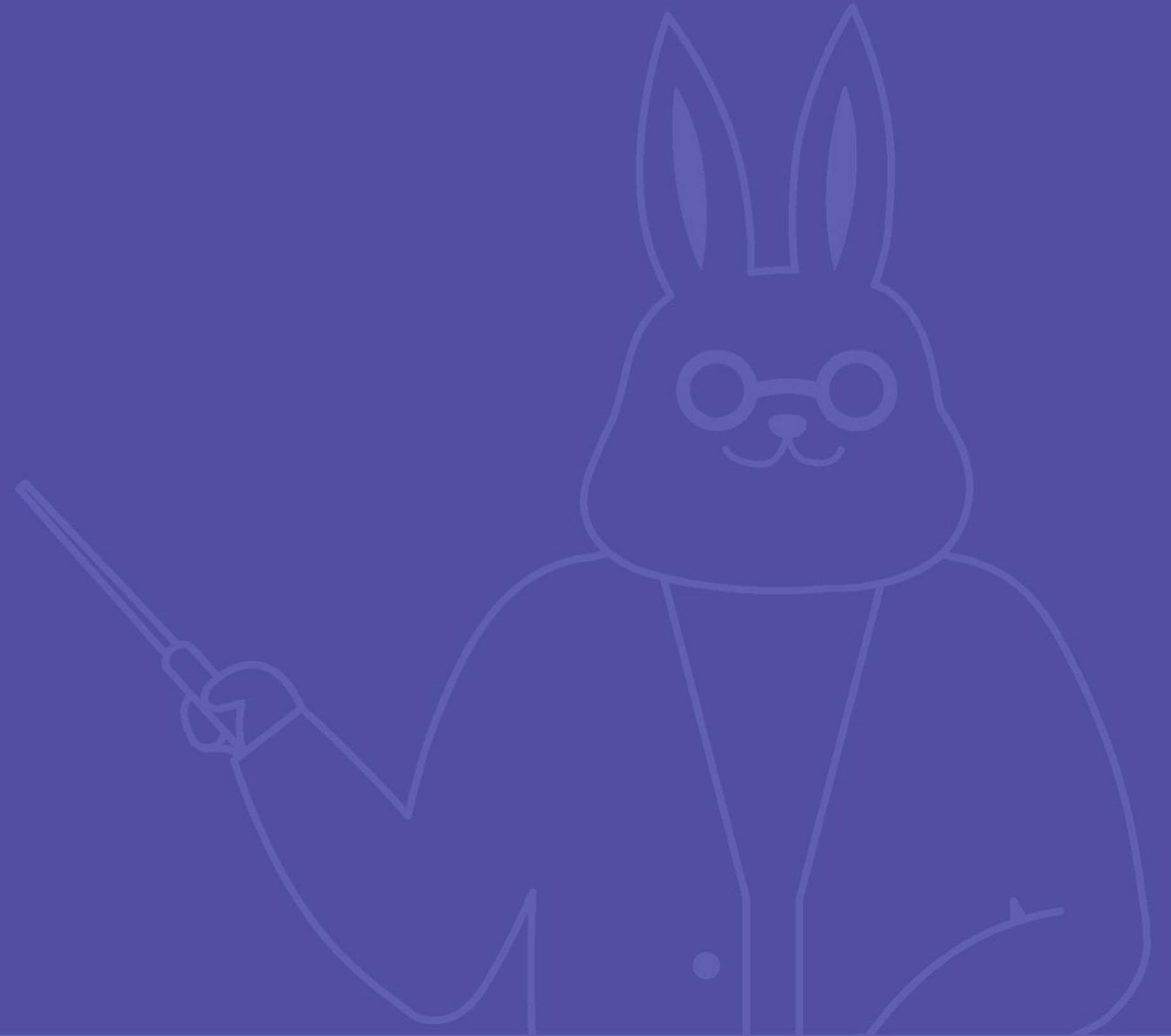
members.name

members.age

db.session.query(Members)
.filter(Members.name == name)

03

Query



✓ CRUD 예제 – Create : 데이터를 저장해요

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

```
members = Members() → members.id = 1 → db.session.add(members)  
                        members.name = 'elice' → db.session.commit()  
                        members.age = 20
```

✓ CRUD 예제 – Read : 데이터를 읽어오

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

```
members = db.session.query(Members)  
    .filter(Members.name == 'elice').all()
```

✓ CRUD 예제 – Update : 데이터를 수정해요

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

```
db.session.query(Members)  
.filter(Members.name == 'elice').all()
```

→

```
members.name = 'oliver'
```

↓

```
db.session.commit()
```


✓ CRUD 예제 – Delete : 데이터를 삭제해요

models.py

```
class Members(db.Model):  
    __tablename__ = 'my_user'  
    id = db.Column(db.Integer, primary_key=True, nullable=False)  
    name = db.Column(db.String(20), nullable=False)  
    age = db.Column(db.Integer, nullable=False)
```

```
me = db.session.query(Members)  
    .filter(Members.name == 'elice').first()
```

→

```
db.session.delete(me)
```

↓

```
db.session.commit()
```

✓ Query 사용법 – equal

Example

```
@app.route('/')
def list():
    member_list = Member.query.filter(Member.name == 'Elice')
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – not equal

Example

```
@app.route('/')
def list():
    member_list = Member.query.filter(Member.name != 'Elice')
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – like

Example

```
@app.route('/')  
def list():  
    member_list = Member.query.filter(Member.name.like('Elice'))  
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – in

Example

```
@app.route('/')
def list():
    member_list = Member.query.filter(Member.name.in_(['Elice', 'Dodo']))
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – not in

Example

```
@app.route('/')
def list():
    member_list = Member.query.filter(~Member.name.in_(['Elice', 'Dodo']))
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – is null

Example

```
@app.route('/')  
def list():  
    member_list = Member.query.filter(Member.name == None)  
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – is not null

Example

```
@app.route('/')  
def list():  
    member_list = Member.query.filter(Member.name != None)  
    return " ".join(i.name for i in member_list)
```


✓ Query 사용법 – and

Example

```
from sqlalchemy import and_
@app.route('/')
def list():
    member_list = Member.query.filter(and_(Member.name == 'Elice',
                                            Member.age == '15'))
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – or

Example

```
from sqlalchemy import or_
@app.route('/')
def list():
    member_list = Member.query.filter(or_(Member.name == 'Elice',
                                           Member.age == '15'))
    return " ".join(i.name for i in member_list)
```

✓ Query 사용법 – order by

Example

```
@app.route('/')
def list():
    member_list = Member.query.order_by(Member.age.desc())
    return " ".join(i.name for i in member_list)
```

Query 문 실행 결과 오름차순으로 정렬(desc() 사용시 내림차순)

✓ Query 사용법 – limit

Example

```
@app.route('/')
def list(limit_num = 5):
    if limit_num is None:
        limit_num = 5
    member_list = Member.query.order_by(Member.age.desc()).limit(limit_num)
    return " ".join(i.name for i in member_list)
```

Query 문 실행 결과에서 limit 크기만큼 반환

✓ Query 사용법 – offset

Example

```
@app.route('/')
def list(off_set = 5):
    if off_set is None:
        off_set = 5
    member_list = Member.query.order_by(Member.age.desc()).offset(off_set)
    return " ".join(i.name for i in member_list)
```

Query 문 실행 결과에서 offset 크기만큼 앞에서부터 생략하고 반환

✓ Query 사용법 – count

Example

```
@app.route('/')  
def list():  
    member_list = Member.query.order_by(Member.age.desc()).count()  
    return str(member_list)
```

Query 문 실행 결과로 반환된 tuple 수를 반환

크레딧

/* elice */

코스 매니저

이재성

콘텐츠 제작자

이바울, 이재성

강사

이바울

감수자

이바울

디자이너

강혜정

연락처

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

contact@elice.io

